NETWORK CONTROL SYSTEM

Patent Number: JP61253945
Publication date: 1986-11-11
Inventor(s): TAKAO KOJI
Applicant(s): CANON INC
Requested Patent: JP61253945
Application Number: JP19850095366 19850507
Priority Number(s):
IPC Classification: H04L11/00
EC Classification:
Equivalents:

Abstract

PURPOSE:To follow the variation of a network constitution precisely under simple algorithm by starting a network reconstituting means every fixed number of times of reception of a communication right transfer instruction.
CONSTITUTION:A lower node address is set up in a self-node setting address and a token is transmitted by using a lower node address subtracted by '1' as a destination address. If the token is not normally transmitted to the lower node, the self-node decides that the lower node is a communication disabled node, checks whether the lower node address value is the minimum address value fixed on the network or not, and if the value is the minimum address value, tries to transmit the token by using a new lower node address value again subtracted by '1' as the destination address. Thus, the transmission of the token to the node of the address value obtained by successively subtracting '1' from the preceding address is tried and the operation is repeated until the token is received by any one of nodes. Consequently, transmission control devices can be precisely combined under simple algorithm.

⑲日本国特許庁（JP）　⑪特許出願公開

⑫公開特許公報（A）　昭61－253945

⑤1Int.Cl.⁴	識別記号	庁内整理番号	⑬公開　昭和61年(1986)11月11日
H 04 L 11/00	101	S－7830－5K U－7830－5K	

審査請求　未請求　発明の数 1　（全9頁）

⑤4発明の名称　網制御方式

⑳特　願　昭60－95366
㉒出　願　昭60(1985)5月7日

⑦2発明者　高尾　貢司　東京都大田区下丸子3丁目30番2号　キヤノン株式会社内
⑦1出願人　キヤノン株式会社　東京都大田区下丸子3丁目30番2号
⑦4代理人　弁理士　大塚　康徳

－

明　細　書

1．発明の名称

網制御方式

2．特許請求の範囲

複数の伝送制御装置をバス型通信媒体に互いに接続して宛先アドレスを有する通信権委譲命令により通信権を委譲してデータ通信を行なうネットワークシステムの網制御方式であつて、前記伝送制御装置に固有のアドレス値を設定するアドレス設定手段と、該アドレス設定手段の設定値を初期値とし該設定値より一定数を減算して得られた値を前記通信権委譲命令の宛先アドレスとし、該値がネットワークシステムで定めた最小アドレス値より小さい場合にはネットワークシステムの定めた最大アドレス値より一定値を減算して得られたアドレス値を固有のアドレス値とする伝送制御装置に対して通信権委譲命令の伝送を試みるネットワーク再構成手段とを備え、前記伝送制御装置は前記通信権委譲命令を一定回数受け取る毎にネットワーク再構成手段を起動することを特徴とする網制御方式。

3．発明の詳細な説明

〔技術分野〕

本発明は複数の伝送制御装置をバス型通信媒体に接してなるネットワークシステムの網制御方式に関するものである。

〔従来技術〕

近年、一本の伝送路を共用して多数の通信制御装置（以下ノードと称す）を接続し、これらノード間でデータ通信を行なうローカルエリアネットワークシステム、いわゆる"ＬＡＮ"が盛んである。ＬＡＮにおける種々の通信方式の中で、トークンパッシング方式はネットワークが高トラヒック状態でも均等に全ノードに通信サービスを行なうことができ、それ故伝送効率の低下を招かないという点で特にすぐれたものである。

このＬＡＮのシステム構成例を第１図に示す。

図中１は伝送路１００～１６０はＬＡＮを構成する各ノードＡ～Ｇである。各ノードＡ～Ｇ（１００～１６０）にはそれぞれ伝送情報の処理を行なうホストＡ～Ｇ（２００～２６０）が接続されている。このように一般にバスＬＡＮでは一本の伝送路を共用している為、各ノードでの発信を無管理で許すと、同時に２つ以上のノードから送信が行なわれる状態が起こり、伝送路上のデータが混信するいわゆる"衝突"現象が生じる。

これを防ぐための一つの方式としてトークンパッシング方式がある。トークンパッシング方式ではこれに対し"トークン"なる送信権委譲命令を伝送路を介して、各ノード間で交換し合い、トークンを受信したノードのみが発信する権利（送信権）を獲得する取極としておき、このトークンを受信したノードは必要があれば（送信要求があれば）送信処理等の一連の通信処理をその時点で行い、この処理が終了した時や、あるいは全く送信処理の必要がない時は、次にトークンを委譲すべきノードにトークンを渡す（送信する）という方法をとつて衝突現象を防いでいる。

"トークン"は、この様にして送信要求のあるノードでの送信処理が終了する毎に次のノードに渡されてネットワーク内のノード間に順次巡回されてゆき、ネットワーク内の各構成ノードでは均等に送信権獲得の機会、即ち通信サービスの機会が与えられるわけである。

トークンの巡回制御は、ネットワークの形状と関連して"トークンリング"方式と"トークンバス"方式の２つの方式があるが、トークンバス方式においては、通常、伝送路を単一の線状導体で形成しており、１つのノードより送信された伝送データは、殆ど同時にネットワークの伝送路（第１図伝送路１）に接続された他の全てのノードに受信される。そこで前述のトークン命令も含めて伝送するデータの先頭には必ず相手先ノードアドレスが附されており、これを各ノードに固有に割当てられたアドレス値と比較して、自ノードに割当てられたノードアドレスと一致した場合に、その伝送データを自ノード宛のデータとして取り込むか、あるいは無差別に伝送データを取り込んだ後で、上記アドレス比較を行ない、一致した時初めて正式の自ノード宛受信データとして扱う。

このトークンバス方式においては、あるノードが次にトークンを渡すべきノード（下流ノード）とは、一般に「自ノードアドレスに最も近くかつ

小なるアドレス値をもつノード。但し、最少値アドレスのノードに対しては、最大のアドレス値をもつノード」と定義されている。

さて、ネットワークシステムは当然の事ながら、故障により動作不可能となる部分が発生する。これはいかなる高価、高信頼性の部品を使用したとしても免れ得ない事である。そしてこの一部の故障がシステム全体の動作に影響を及ぼす事態が生じた時、大きな問題となる。

ネットワークシステムを構成する１つのノードがダウンすると、トークンがこのノードで消滅してしまい次のノードにトークンが送られなくなる。このため、このままではトークンの巡回が行なわれなくなり、ネットワーク内のいずれのノードに於いても通信不可となる重大事態が生じる。

以前においては、障害によるこの様なシステムのダウンが容認される事もあつたが、近年においては、故障に対する体力（フォルト・トレランス）がシステムの信頼性の目安の一つとして重要視されており、この耐力を強化する為に、例えば故障部分をシステムから切り離し、残りの部分で動作を続行するいわゆる“縮退運転”や、動作中に故障から回復した部分の再組み込みを行なうといつた手段が必要とされる。

更にネットワークに於ては故障ではないにもかかわらず、動作不可能なノードが存在する事を拒めない事情がある。即ち、ネットワークではこれに接続される機器やオフイスの配置が広範囲に分散する為、ネットワーク内のノードを一斉に立上げるよりは部分的に立上げていけた方がユーザーにとつて都合がよく、また、業務の終了した機器のノードは個々にパワーダウンできる方が好都合である。これらの場合、一部に動作不可能ノードを含みながらネットワークの運転を行なわしめる事になる。

以上の事を考えると、ネットワーク内の動作可能、不可能なノードの構成変動は常に生じ得るとすべきであり、これに対従してネットワーク内の各ノードでは、次にトークンを渡すべき下流ノードの状態は常に監視し、もし、下流ノードが動作不可能に陥いつたと判明すれば、ただちにこのノードをネットワーク構成より外し、トークンの巡回リングより外し（即ち縮退運転し）、トークンを以後渡すべきノードとして別の適当なノードを選別し、これに切り換える事が必要となる。

第１図に示すシステムにおいて例えばノードＢ１１０とノードＦ１５０が動作不可能ノードである場合のトークンの巡回例を第３図に示す。

図示の如く動作不可ノードを避けながらトークンを巡回させる事が必要である。

更に、今までトークンを渡していたノードと自ノードとの中間のアドレス値をもつノードが新たに動作可能状態となり、ネットワークへの参入（トークンを受る）の準備が整つた場合には、トークンを渡すべき下流ノードをこの新しいノードに切り換える事が必要である。

以上の動作を前述したトークンを渡すべきノードの定義として言い直せば

「“動作可能な状態の”ノードのうち、自ノードアドレスに最も近く、かつ小なるノード。但し、該当するノードがない時は、“動作可能な”ノードのうち、最大値アドレスをもつノード」という事になる。

さて、以上述べた様なネットワーク内のノード

の構成の変動に対しては従来では十分に追従できるものが少なく、従つて例えば一部のノードのダウンがネットワーク全体のダウンにつながるというケースが多かつた。

また、上記構成変動に対処する手段を有するものであつても、その処理手順は非常に複雑であり、特に通信制御ファームウエアのオーバーヘッドが大きい事から、シンプルなアルゴリズムが求められている。

〔目的〕

本発明は上述した従来技術の欠点に鑑みなされたものでネットワーク構成の変動に確実に追従でき、かつ、これをシンプルなアルゴリズムで行なうことのできる網制御方式を提供する事を目的とする。

〔実施例〕

以下、図面を参照して本発明の一実施例を詳細に説明する。

第４図は本発明の一実施例のノードのブロックである。図中１は第１図に示したＬＡＮのネットワーク伝送路、２は伝送制御装置であるノード、３はノード２に接続される各種コンピュータ機器や事務機器（第１図図示のホスト２３０～２６０に対応）である。

ノード２は伝送路１と間のでデータ通信を行う送／受信回路４、ノードの全体制御及び発信データの加工や受信データの解読、分解や通信動作のタイミング制御等を行なう制御部（以下ＣＰＵと称す）５、送受信データ等を蓄積するメモリ回路６、そしてホスト３との間のインターフエース回路７、各ノードに固有のアドレス番号を設定するスイツチ等で構成されるアドレス設定部８より構成されている。アドレス設定部８に設定されたアドレス値はＣＰＵ５により読み出され、送受信時の宛先アドレス、また送信元アドレスとして利用される。

ホスト３よりの送信すべきデータは、一旦ノード２内のメモリ回路６蓄えられ、通信データとしての適切なるフオーマツト化（パケツト化）がなされ、送信先ノードアドレスを宛先アドレス値、アドレス設定回路８の設定値を送信元アドレスとして付加した後、自ノード宛トークンを受信し、送信権を獲得した時に伝送路１に送出する。

一方、他ノードでは、伝送路１上の通信データを受信し、受信データ中の宛先アドレス値を調べ、自ノード内のアドレス設定回路８の設定値、即ち自ノード宛のデータであると判断すると、このデータを自ノード内に取り込み、ＣＰＵ５で多少の分解、編集処理を行なつた後、接続されたホスト３へと配送する。

また、トークンパツシング方式では、伝送路１上を流れるデータは全てホスト３相互間での通信データばかりではなく、トークン等の各ノード内のＣＰＵ自身と、相手先ノードのＣＰＵとの間のみのいわゆる通信制御データも含まれる。

以上のハードウエア構成と前述トークンパツシングの原理に基づき、ネットワーク内の各々のノードで、トークンを次々に下流ノードに受け渡していく事で一本の伝送路を共用した通信がなされるわけである。

さて、各ノードにおいては、一般にトークンを渡すべき下流ノード（即ち、トークンを送信する際に付加する宛先アドレスで指定するノード）は、一旦、トークンがネットワーク上を巡回して

しまうと、固定化されるケースが多い。

さて、各ノードにおいては、一般にトークンを渡すべき下流ノード（即ち、トークンを送信する際に付加する宛先アドレスで指定するノード）は、一旦、トークンがネットワーク上を巡回してしまうと、固定化されるケースが多い。

これに対し、本実施例においては以下に述べるRECONFIGURATION（再構成）動作を各ノードに義務付け、ネットワーク構成の変動に対応して、柔軟に、下流ノードを変えていく方法をとる。

以下、本実施例のデータ伝送制御を第５図のフローチャートを参照して説明する。

まずステップＳ１０で伝送路１上の伝送データを監視し、トークンが受信されたか否かを調べる。トークンが受信されない時はステップＳ１１に進み、宛先アドレスが自ノードの設定アドレス値であり、自ノード宛の伝送データか否かを調べ、自ノード宛の伝送データであればステップＳ１２でこのデータを受信し、必要に応じてホスト３にこの受信データを送る。

トークンを受信したときはステップＳ１０よりステップＳ１３に進み、ホスト３より送信すべきデータがあり、送信要求があるか否かを調べ、送信要求があればステップＳ１４でデータ送信処理を実行しステップＳ１５に進む。ステップＳ１３で送信要求がないときはそのままステップＳ１５に進む。

ステップＳ１５では受信したトークンが自ノードが電源投入後の最初のトークンか否かを調べ、最初のトークンのときはステップＳ２１に進む。これはそのノードが伝送動作を開始（例えばパワーオン）して初めてトークンを受け取つた後のトークン委譲動作時を表し、立上つたばかりで、自分の受け取つたトークンを受け渡すべき下流ノードをまだ認識していないことを意味し、これに対処するためである。

ステップＳ１５で最初のトクーンでない場合にはステップＳ１６に進み、ステップＳ１６で障害パワーダウン等不測の事態により、それまで巡回していたトークンが消失し、続くリカバリー処理を行なつた後の最初のトークン委譲動作か否かを調べ、最初のトークン委譲の場合にはステップＳ２１に進む。これはトークンにより通信権を委譲しているシステムにおいてはよく発生する事であり、例えば、丁度トークンが巡回してきたノードがパワーダウンされた場合、そこでトークンが消失してしまいトークンを後のノードに受け渡す事ができなくなるためである。

この様な場合、いずれかのノードでトークンを発生し、トークンの巡回を再開するリカバリー処理を行う必要があるからである。

ステップＳ１６でトークン消失後の最初のトークン受信でない場合にはステップＳ１７に進み、トークンがそのノードに巡回してくるある一定周期ごとのトークン委譲動作時、もしくは、一定時間周期ごとのトークン委譲動作時か否かを調べ、一定周期経過時の場合にはステップＳ２１に進む。これはネットワークに新規に参入するノードに対処する為である。即ち、今までの下流ノードより更に自ノードに近いアドレス値のノードが立上り、動作可能となつたケースでこの新規参入ノードの前後のノードがこれを知らずにいる場合、いつまでたつてもトークンはこの新規参入ノードにはまわされてこない。そこである一定周

期で後述するRECONFIGURATION 動作を行なわせ、これら新規参入ノードを認識し、ネットワークに参入させるためである。

ステップＳ１７で一定周期経過していない場合にはステップＳ１８に進み、メモリ回路６内の不図示の下流ノードアドレス値を宛先アドレスとしてトークンを送信する。そして続くステップＳ１９でトークンの委譲が正常に行われたか否かを調べる。これを識別する手段は各ネットワークの伝送制御手順によつて様々に異なるが、例えばトークンを受け取つたノードで肯定応答であるＡＣＫ応答を返すとか、あるいは受信ノードが新たな通信動作を開始し、次に伝送路１に送出された通信データの送信元アドレスがこの下流ノードアドレスであることを判別することにより行われる。

トークンの委譲が正常に行われた場合には再びステップＳ１０に戻り、次のデータ伝送に備える。

ステップＳ１９でトークンの委譲が正常に行われなかつた場合にはステップＳ２０でトークンの送信が２度目か否かを調べ、１度目の場合にはステップＳ１８に進み、トークンの再送信処理を実行する。ステップＳ２０でトークンが２度行われた場合にはステップＳ２１に進む。これは今までトークンを渡していた相手ノードが不意にダウンした（故障あるいは電源 OFF）場合に対処するものであり、トークンを委譲すべき新しいノードを捜し出すためである。

ステップＳ２１では新たにトークン委譲先を検出するネットワーク再構築処理であるRECONFIGURATION 動作を実行し、トークンを正常に委譲し、ステップＳ１０に戻る。

ステップＳ２１のRECONFIGURATION 動作の詳細を第６図のフローチャートを参照して以下に説明する。

まずステップＳ１で今までの下流ノードアドレスとは無関係に、下流ノードアドレスを自ノードの設定アドレスに設定し、続くステップＳ２でこの下流ノードアドレスを１だけ減算し、ステップＳ３でこの下流ノードアドレスを宛先アドレスとしてトークンを送信する。そして続くステップＳ４でトークンの受け渡しが成功したか否かを調べる。これを識別する手段はステップＳ１９と同様である。

トークンが下流ノードに正常に送られなかつた場合（送信が失敗すれば）には、相手ノードは動作（通信）不可能なノードであるとしてステップＳ６に進み、下流ノードアドレス値がネットワークに定める最小アドレス値か否かを調べ、最小アドレス値である場合にはステップＳ２に戻り、再び下流ノードアドレス値を１つ減算し、この新たな下流ノードアドレス値を宛先アドレスとしてトークン送信を試みる。

こうして、アドレス値を順次減算して行つてはそのアドレス値のノードにトークンの送信を試み、いずれかのノードにトークンが受信されるまで、この動作を繰り返す。

もし、上記減算されていつたアドレス値がネットワークに定める最小アドレス値に達しても、上記トークン受け渡しが成功しない場合には、ステップＳ５よりステップＳ７に進み、下流ノードアドレス値をネットワークに定める最大ノードアドレス値としてステップＳ３に戻る。そして以後は

また、トークンの受け渡しとアドレス値の減算動作を繰り返す。

もし、トークンの受け渡しが成功した場合には、ステップＳ４よりステップＳ７に進み、そこで本動作をやめ、その時の下流ノードアドレス値を記憶し、以後その時のアドレス値を下流ノードアドレスとする。

RECONFIGURATION 動作の処理は時間を要する為、ステップＳ１５，１６でＹＥＳ，ステップＳ２０でＮＯの如き特定ケースとステップＳ１７のＹＥＳの如く一定周期毎にのみ行なわれる様にし、他の通常のトークン委譲時に際しては以前のRECONFIGURATION 動作で求めた次ノードアドレス値を固定値として、そのまま用いる事でネットワークのデータ伝送効率を低下させない様に配慮している。

以上説明したように本実施例によればトークンの委譲がどのような場合においても確実に行え、かつ、トークン委譲が正常に行えなかつた場合にもデータ伝送の効率をほとんど損なうことなくネットワークの再構築を行える。

〔効果〕

以上説明した様に本発明によれば、ネットワークシステムにおいて、故障その他で動作不可能となる伝送制御装置の切り離しや、新たに動作可能となつた伝送制御装置の組み込みを確実に、しかもシンプルなアルゴリズムで行うことができる。

４．図面の簡単な説明

第１図はバス型ネットワークシステム構成を示す図、

第２図、第３図はトークンバス方式におけるトークン巡回の様子を示す概念図である。

第４図は本発明に係る一実施例のブロック図、

第５図は本実施例のデータ伝送制御を示すフローチャート、

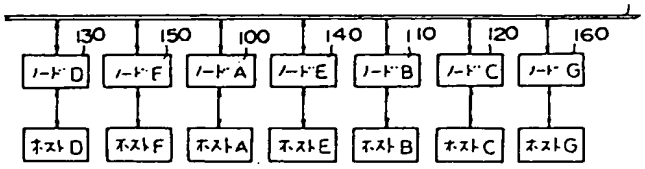
第６図は本実施例のネットワーク再構成動作フローチャートである。

ここで、１…ネットワーク伝送路、２…ノード、３…ホスト、４…送／受信回路、５…ＣＰＵ、６…メモリ回路、７…インタフエース回路、８…アドレス設定回路である。

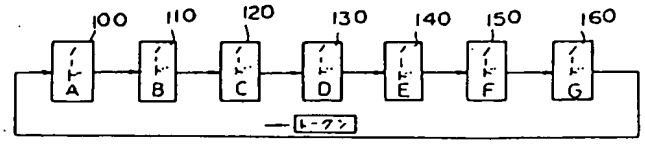
特許出願人　　キヤノン株式会社

代理人　弁理士　　大塚康徳

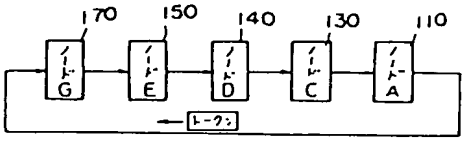
第１図



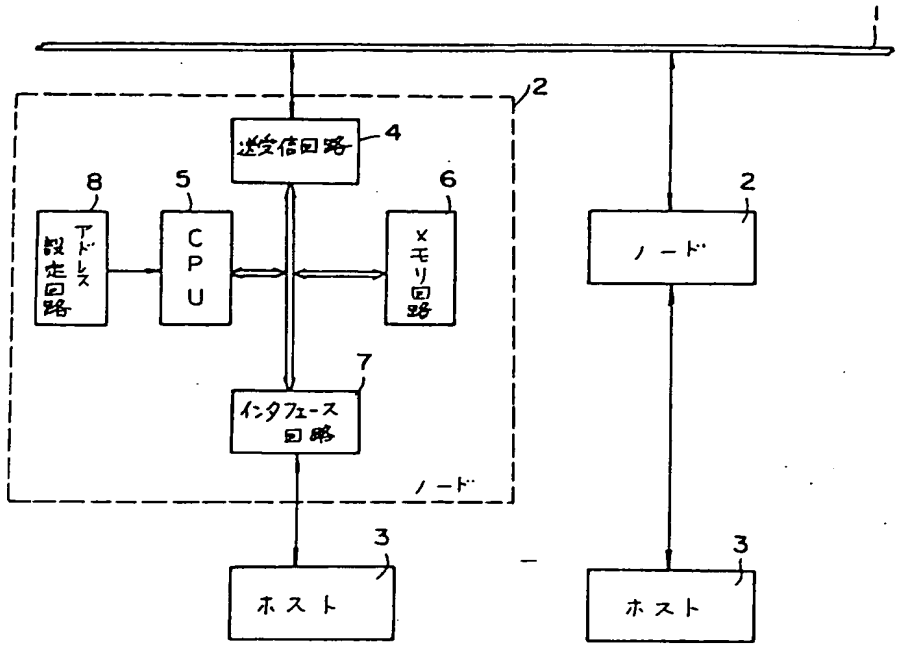
第２図



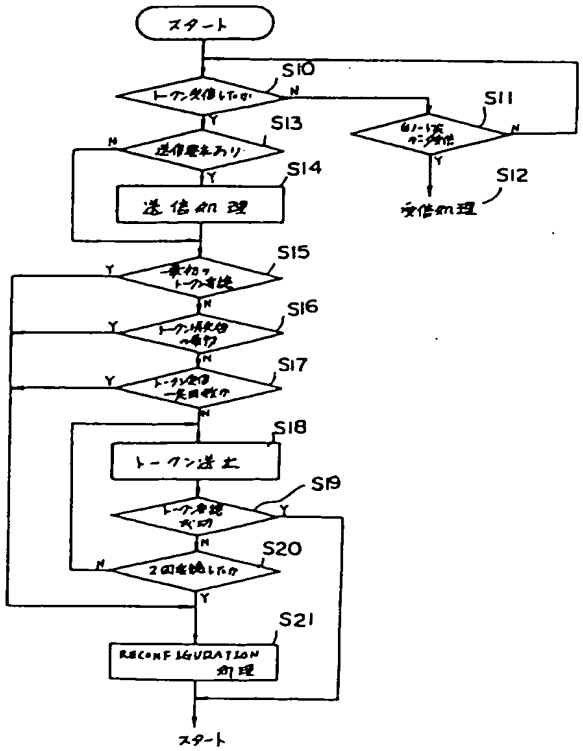
第３図



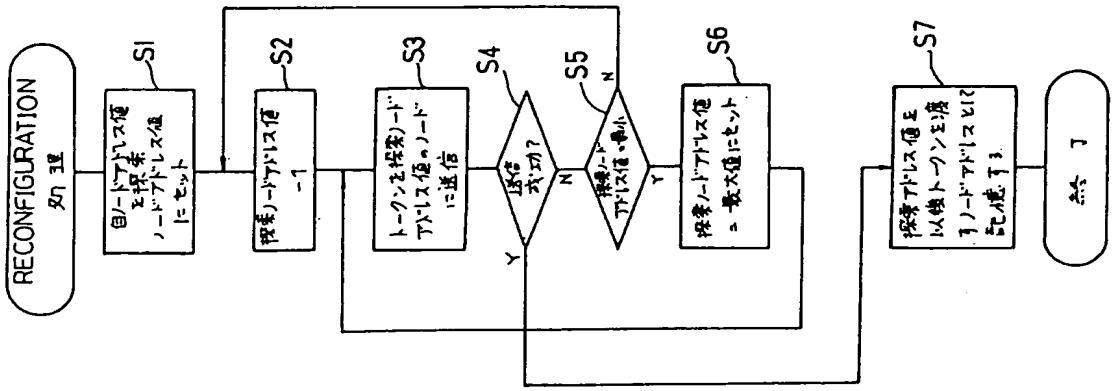
第 4 図


1
2
送受信回路
4
8
アドレス設定回路
5
CPU
6
メモリ回路
7
インタフェース回路
ノード
3
ホスト
2
ノード
3
ホスト

第 5 図


スタート
S10
S11
S12
S13
S14
送信処理
S15
S16
S17
S18
トークン送出
S19
S20
S21
RECONFIGURATION 処理
スタート

第 6 図



PTO 04-0146

Japanese Patent

Document No. S61-253945

Network Control System

Mo Seigyo Hoshiki

Koji Takao

UNITED STATES PATENT AND TRADEMARK OFFICE

Washington, D.C. October 2003

Translated by: Schreiber Translations, Inc.

Country	:	JP
Document No.	:	S61-253945
Document Type	:	A
Language	:	Japanese
Inventor	:	Koji Takao
Applicant	:	Canon Inc.
IPC	:	H04L 11/00
Application Date	:	May 7, 1985
Publication Date	:	November 11, 1986
Foreign Language Title	:	Mo Seigyo Hoshiki
English Title	:	Network Control System

/1[1]

SPECIFICATION

1. Title of the Invention

Network Control System

2. Claims

A control system of a network system in which plural transmission control devices are mutually connected to a bus-type communication medium and data communication is performed by transferring a communication right by a communication right transfer instruction having a destination address, characterized in that it comprises an address setting means which sets an address value specific to said transmission control device; and a network reconfiguring means which sets a value set by that address setting means to an initial value and takes a value obtained by subtracting a fixed number from that set value as a destination address for said communication right transfer instruction, and when that value is smaller than a minimum address value defined in the network system, attempts transmission of the communication right transfer instruction to a transmission control device taking a value of an address obtained by subtracting a fixed number from a maximum address value defined in the network system as the specific value; and

[1] Numbers in the margin indicate pagination in the foreign text.

said transmission control device starts up the network reconfiguring means each time it receives said communication right transfer instruction a fixed number of times.

/2

3. Detailed Explanation of the Invention

[Technological Field]

The present invention relates to a network control system of a network composed of plural transmission control devices being connected to a bus-type communication medium.

[Prior Art]

Recently, the so-called "LAN," or local area network, in which a large number of communication control devices (below called nodes) are connected sharing one communication path and data communication is performed among these nodes, has come to flourish. Among the various communication systems in LANs, token passing systems are particularly excellent in that communication services can be performed equally for all nodes even in a state of high network traffic, and therefore it does not lead to degradation of transmission efficiency.

Fig. 1 shows an example of the system configuration of this LAN. In the drawing, 1 is a communication path, and 100-160 are respective nodes A-G constituting the LAN. To each node A-G (100-160), a host A-G (200-260) which performs processing of

transmitted information is connected. Thus, because one communication path generally is shared in a bus LAN, if transmission at each node is allowed without control, a state in which transmission is performed from two or more nodes simultaneously is caused, and the so-called "collision" phenomenon in which data is mixed on the communication path is produced.

As one system for preventing this, there is the token passing system. In the token passing system, in contrast, the collision phenomenon is prevented by adopting a method whereby a transmission right transfer instruction serving as a "token" is exchanged mutually among the respective nodes over the communication path as an arrangement such that only the node receiving the token acquires the right to transmit (transmission right), the node receiving this token, if necessary (if there is a transmission request), performs a series of communication processing such as transmission processing at that time, and when this processing is finished, or when there is no need at all for transmission processing, next the token is passed (transmitted) to the node to which the token should be transferred.

The "token" thus is passed to the next node each time transmission processing at a node for which there is a

transmission request is finished, it goes on to be circulated in sequence among the nodes in the network, an opportunity for acquisition of transmission right, that is, an opportunity for communication service, is provided equally among all the constituent nodes in the network.

For control of the circulation of the token, there are the "token ring" system and the "token bus" system, relating to the shape of the network, but in the token bus system, usually, the communication path is constituted by a single linear conductor, transmission data transmitted from one node is received almost simultaneously by all the other nodes connected to the communication path (communication path 1 in Fig. 1) of the network. Therefore, the address of the destination node necessarily is attached in the header of the transmitted data including also said token instruction, this is compared with the address value allocated specifically to each node, and when it matches with the node address allocated to the self node, that transmission data is taken in as data addressed to the self node, or, the above address comparison is performed after the transmission data was taken in indiscriminately, and it is first handled as reception data addressed to the self node when it matches.

In this token bus system, a node to which a certain node next should pass the token (downstream node) generally is defined as "the node having an address value

/3

closest to and smaller than the self node address, however, for the node having the minimum value address, the node having the maximum address value."

Now, a network system of course comes to have parts that become incapable of operation due to failure. This is an unavoidable fact even when using expensive, highly reliable components. Also, when a situation arises in which the failure of this part influences the operation of the overall system, then it becomes a big problem.

If one node constituting the network system goes down, the token vanishes at this node, and the token no longer can be sent to the next node. Because of this, circulation of the token is no longer performed in this condition, and a serious situation arises in which all the nodes in the network become incapable of communication.

In the above, while such system downtime due to failure has been accepted, recently, resistance <typo in source> to failure (fault tolerance) is being considered important as one measure of system reliability, and in order to strengthen this

resistance, for example, so-called "degenerate operation" in which a failed part from the system is cut off and operation is continued at the other parts, and a means for performing reincorporation of a part restored from failure during operation, are considered necessary.

Furthermore, the fact that nodes incapable of operation are present despite not having failed in a network cannot be denied. That is, in a network, because the devices connected to this and the placement of the offices are distributed over a wide area, rather than starting up the nodes in the network all together, it is more convenient for the user to go on to start them up in parts. Also, it is convenient that the nodes of devices having finished the tasks be capable of powering down individually. In these cases, it becomes that operation of the network is accomplished while partially including nodes incapable of operation.

Considering the above matters, it should be assumed that variation of the configuration of nodes capable and incapable of operation in a network should is always possible to happen, and in response to this, at each node in the network, the state of the downstream node to which the token next should be passed is regularly monitored, and if it is determined that the downstream node has become incapable of operation, it becomes necessary to

remove this node immediately from the network configuration, to remove it from the token circulation ring (that is, degenerate operation is performed), to select another suitable node as the node to which the token next should be passed, and to change to this.

Fig. 3 shows an example of token circulation in the case that, for example, node B110 and node F150 are nodes incapable of operation in the system shown in Fig. 1.

As illustrated, it is necessary to circulate the token while avoiding the nodes that are incapable of operation.

Furthermore, in case that a node having an intermediate address value between the nodes to which the token was passed up to now and the self node has become newly in a state capable of operation, and preparation was made for participation in the network (receiving the token), it is necessary to change the downstream node to which the token should be passed to this new node.

To restate the above operation as the definition of the node to which the token should be passed previously described, it becomes:

"the node being closest and smaller than the self node address among the nodes 'in a state capable of operation', however, when

there is no such node, the node having the maximum value address among the nodes 'capable of operation'."

Now, in response to variation of the configuration of

/4

nodes in a network as described above, in the past, there were few which were sufficiently capable of tracking. Accordingly, there were many cases in which, for example, the downtime of some of the nodes was connected to the downtime of the entire network.

Also, even if it is one having a means for handling the above variation of configuration, because the processing means is very complex, and particularly the overhead for firming down the communication control is great, a simple algorithm has been sought.

[Purpose]

The present invention was created in consideration of the drawbacks of the prior art described above, and its purpose is to provide a network control system in which variation of a network configuration can be tracked assuredly, and this can be performed with a simple algorithm.

[Working Examples]

Below, a working example of the present invention is explained while referring to the drawings.

Fig. 4 is a block drawing of the nodes of one working example of the present invention. In the drawing, 1 is the network transmission path of the LAN shown in Fig. 1, 2 is a node which is a transmission control device, and 3 is some type of computer or office machine (corresponds to host 230-260 shown in Fig. 1) connected to the node 2.

The node 2 is constituted by a transceiver circuit 4 which performs data communication over the transmission path 1, a controller (below called CPU) 5 which performs overall control of the node as well as processing of transmission data, analysis and decomposition of reception data, timing control of communication operations, and the like, a memory circuit 6 which stores transmission and reception data 6, <spurious dash> an interface 7 with the host 3, and an address setting part 8 which is constituted by a switch, and the like, for setting an address number specific to each node. The address value set by the address setting part 8 is read out by the CPU 5, and it is used as the destination address and also as the transmission source address during transmission and reception.

Once data to be transmitted from the host 3 is stored in the memory circuit 6 in the node 2, suitable formatting (packetizing) as communication data is performed, and after adding the transmission destination node address as the

destination address value and the value set by the address setting circuit 8 as the transmission source address, it is sent out to the transmission path 1 when a token addressed to the self node is received and the transmission right is acquired.

On the other hand, at another node, communication data on the transmission path 1 is received, the destination address value in the reception data is examined, and if it is determined to be the value set by the address setting circuit 8 in the self node, that is, it is data addressed to the self node, this data is taken into the self node, and after performing some amount of decomposition and editing processing in the CPU 5, it is delivered to the connected host 3.

Also, in the token passing system, the data flowing on the transmission path 1 is not all only communication data among the hosts 3, and so-called communication control data, which is only between the CPU itself inside each node and the CPU of the destination node, such as the token, also is included.

Based on the above hardware configuration and the principle of said token passing, by receiving the token and passing it to the downstream node at each node in the network, communication that shares one transmission path is accomplished.

Now, there are many cases in which at each node, the downstream node to which the token should be passed (that is,

the node designated by the destination address added when transmitting the token) generally is made fixed

/5

once the token finishes circulation on the network.

<Last paragraph repeated word for word>

As opposed to this, in the present working example, a method is adopted in which each node is tasked with a RECONFIGURATION (reconfiguration) operation described below, and the downstream node goes on to be changed flexibly in response to variation of the network configuration

Below, the data transmission control of the present working example is explained while referring to the flow chart in Fig. 5.

First, in step S10, the transmission data on the transmission path 1 is monitored, and it is examined as to whether or not a token was received. When a token is not received, the flow advances to step S11, and it is examined as to whether or not the destination address is the set address value of the self node and it is transmission data addressed to the self node. If it is transmission data addressed to the self node, this data is received in step S12, and this received data is sent to the host 3 according to need.

When a token was received, the flow advances from step S10 to step S13, and it is examined as to whether or not there is data to be transmitted from the host 3 and there is a transmission request. If there is a transmission request, data transmission processing is executed in step S14, and the flow advances to step S15. When there is no transmission request in step S13, the flow advances as such to step S15.

In step S15, it is examined as to whether or not the received token is the first token after the self node was powered on, and if it is the first token, the flow advances to step S21. This indicates the time of token transfer operation after that node has begun transmission operation (for example power on) and has first received a token, it means that it just was brought up and it still does not recognize the downstream node to which the token received by itself should be passed, and it is for the purpose of dealing with this.

If it is not the first token in step S15, the flow advances to step S16, and in step S16, it is examined as to whether or not the token having circulated up to now was lost due to an unexpected situation such as failure power down and it is the first token transfer operation after having performed subsequent recovery processing. If it is the first token transfer, the flow advances to step S21. This is because it often occurs in

systems in which the communication right is transferred by token, for example, if the node is powered down just when the token was circulated to it, the token gets lost there and it is no longer possible to pass the token to later nodes.

It is because, in such case, recovery processing in which a token is generated at any node and circulation of the token is restarted is necessary.

If it is not the first token receipt after loss of the token in step S16, the flow advances to step S17, and it is examined as to whether or not it is a time of token transfer operation for each certain fixed cycle in which the token is circulated up to that node, or a time of token transfer operation for each fixed time cycle. If it is a time that a fixed cycle has passed, the flow advances to step S21. This is for the purpose of dealing with nodes that are newly participating in the network. That is, it is a case in which a node having an address even closer to the self node than the downstream node up to now was brought up and has become capable of operation, and if the nodes before and after this newly participating node are unaware of this, the token will not be circulated up to this newly participating node now matter how much time passes. Therefore, the RECONFIGURATION operation

/6

described later is performed at a certain fixed cycle, and it is for the purpose of recognizing these newly participating nodes and allowing them to participated in the network.

If the fixed cycle has not passed in step S17, the flow advances to step S18, and the token is transmitted with the downstream node address value not illustrated in the memory circuit 6 as the destination address. Also, it is examined as to whether or not the transfer of the token was accomplished normally in the subsequent step S19, and the means for identifying this differs variously according to the transmission control means of each network, but for example, an ACK response which is an affirmative response is returned by the node receiving the token, or the receiving node starts a new communication operation and the fact that the transmission source address of the communication data next transmitted out to the transmission path 1 is the address of this downstream node is discriminated, whereby it is accomplished.

If the transfer of the token was accomplished normally, the flow returns again to step S10, and preparation is made for the next data transmission.

If the transfer of the token was not accomplished normally in step S19, it is determined as to whether or not the transmission of the token was the second time in step S20. If

it was the first time, the flow advances to step S18, and retransmission processing of the token is executed. If the token was performed two times in step S20, the flow advances to step S21. This is for the purpose of dealing with the case in which an open node to which the token was passed up to now has suddenly gone down (failure or power off), and seeking out a new node to which the token should be transferred.

In step S21, the RECONFIGURATION operation, which is network reconfiguration processing for newly detecting token transfer destinations, is executed, the token is transferred normally, and the flow returns to step S10.

The details of the RECONFIGURATION operation of step S21 are explained below while referring to the flow chart in Fig. 6.

First, in step S1, the downstream node address is set to the set address of the self node regardless of the downstream node address up to now, in the next step S2, this downstream node address is decremented by 1, and in step S3, the token is transmitted with this downstream node address as the destination address. Also, in the next step S4, it is examined as to whether or not the transfer of the token was successful. The means for identifying this is the same as in step S19.

If the token was not sent normally to the downstream node (if the transmission failed), the flow advances to step S6 with

the destination node being treated as a node incapable of operation (communication), and it is examined as to whether or not the downstream node address value is the minimum address value defined in the network. If it is the minimum address value, the flow returns to step S2, the downstream node address value is decremented by 1 again, and token transfer is attempted with this new downstream address value as the destination address.

Thus, the address value is sequentially decremented, transmission of the token to the node having that address value is attempted, and this operation is repeated until the token is received by any node.

If the above token transfer does not succeed even if the above decremented address value reaches the minimum address value defined in the network, the flow advances from step S5 to step S7, and it returns to step S3 with the downstream node address value set as the maximum node address value. Also, afterward,

/7

the operation of token transfer and decrementing of the address value are repeated.

If the token transfer was successful, the flow advances from step S4 to step S7, the present operation stops there, the

downstream node address value at that time is stored in memory, and the address value at that time is used as the downstream node address afterward.

Because the processing of the RECONFIGURATION operation takes time, it is made so as to be performed only in special cases such as YES in steps S15 and 16 and NO in step S20, and for each fixed cycle such as YES in step S17, and in the event of the token transfer times of other communications, the self <typo in source> node address value sought in the previous RECONFIGURATION operation is used unchanged as a fixed value, whereby it is placed so as not to lower the data transmission efficiency of the network.

According to the present working example as explained above, token transfer is performed assuredly in whatever case, and even when token transfer could not be performed normally, network reconfiguration can be performed substantially without hindering the efficiency of data transmission.

[Effect]

According to the present invention as explained above, in a network system, cutting off transmission control devices that have failed or otherwise have become incapable of operation and incorporating transmission control devices that have newly

become capable can be performed assuredly and with a simple algorithm.

4. Brief Explanation of the Drawings

Fig. 1 is a drawing showing the configuration of a bus-type network system.

Fig. 2 and Fig. 3 are schematics showing the elements of token circulation in a token bus system.

Fig. 4 is a block drawing showing one working example pertaining to the present invention.

Fig. 5 is a flow chart showing the data transmission control of the present working example.

Fig. 6 is a flow chart of the network reconfiguration operation of the present working example.

Here, 1: network transmission path, 2: node, 3: host, 4: transceiver circuit, 5: CPU, 6: memory circuit, 7: interface circuit, 8: address setting circuit.

Applicant for Patent: Canon Inc.

Agent: Patent Attorney, Yasunori Otsuka <seal>

第 1 図

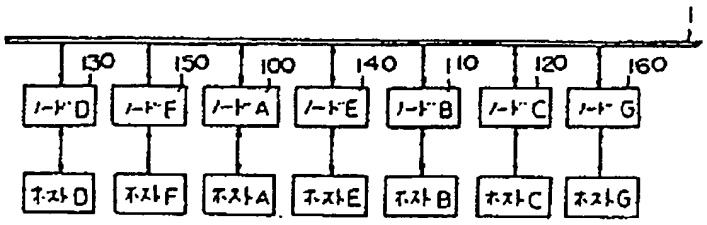


Fig. 1

Node D ... Node G

Host D ... Host G

第 2 図

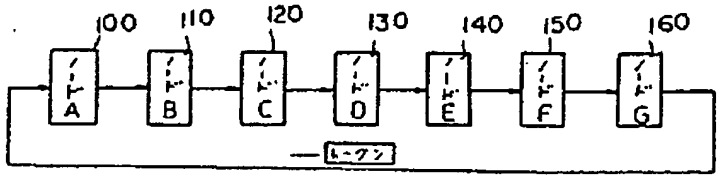


Fig. 2

Node A ... Node G

Token

第 3 図

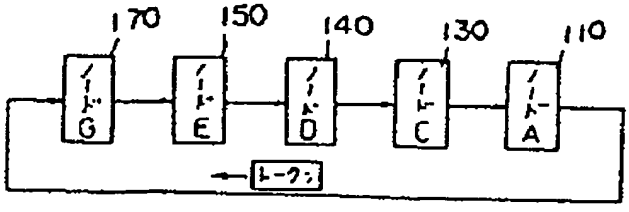


Fig. 3

Node G ... Node A

Token

/8

第４図

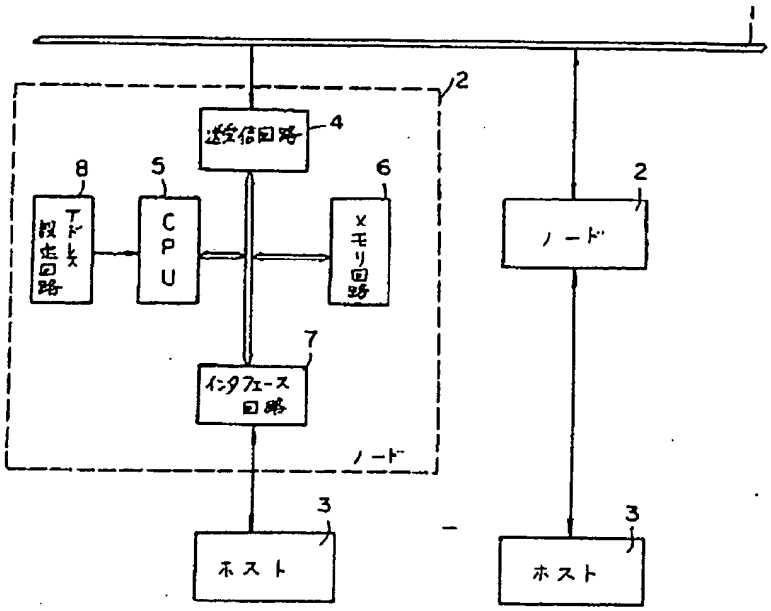


Fig. 4

2 Node

3 Host

4 Transceiver circuit

6 Memory circuit

7 Interface circuit

8 Address setting circuit

第 5 図

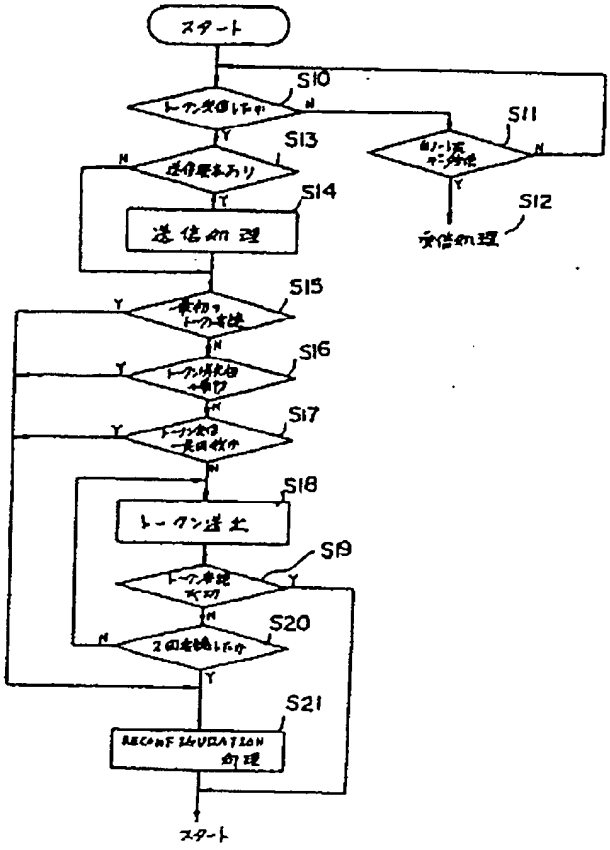


Fig. 5

Start

S10 Is a token received?

S11 Data addressed to self node <illegible>?

S12 Reception processing

S13 Is there a transmission request?

S14 Transmission processing

S15 First token transfer?

S16 Token <illegible>?

S17 Token <illegible> a fixed number of times?

S18 Token transmitted

S19 Token transfer successful?

S20 Transferred 2 times?

S21 RECONFIGURATION processing

Start

第 6 図

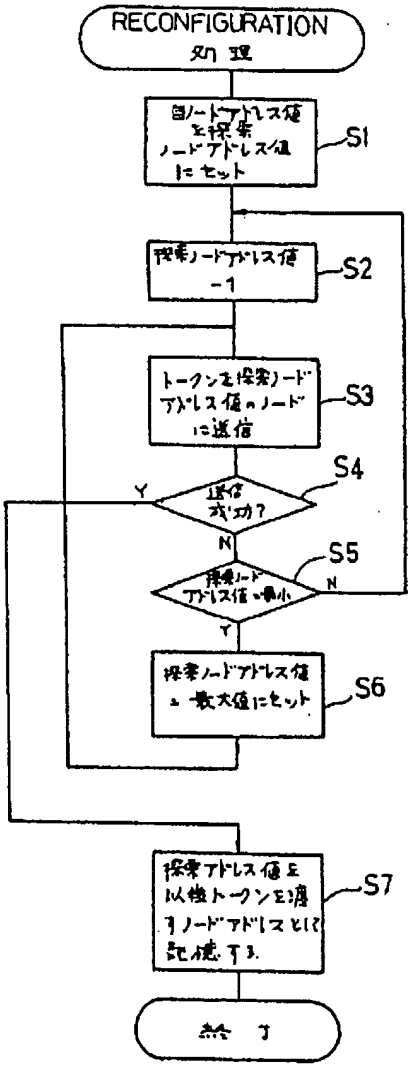


Fig. 6

RECONFIGURATION processing

S1 Set self node address value to probed node address value

S2 Probed node address - 1

S3 Transmit token to node having probed node address value

S4 Transmission successful?

S5 Probed node address value minimum?

S6 Set probed node address value to maximum value

S7 Store probed address value as node address to pass token afterward

End